



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

10

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/954,951	09/18/2001	John R. Hind	RSW920010128US1	8532
7590	12/23/2005		EXAMINER	
Jeanine S. Ray-Yarletts IBM Corporation T81/503 PO Box 12195 Research Triangle Park, NC 27709			STEVENS, ROBERT	
			ART UNIT	PAPER NUMBER
			2176	

DATE MAILED: 12/23/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/954,951	HIND ET AL.
	Examiner Robert M. Stevens	Art Unit 2176

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 15 September 2005.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-53 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-53 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413)
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date. _____.
3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.	5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)
	6) <input type="checkbox"/> Other: _____.

DETAILED ACTION

1. This action is responsive to communications: after final amendment (filed 7/13/2005) in light of pre-appeal brief conference request (filed 9/15/2005) for the application filed by Hind et al entitled "Low-Latency, Incremental Rendering in a Content Framework".
2. The Office withdraws the previous rejections under 35 USC 103(a). This action is non-final.
3. The Office sets forth new rejections of the claims under 35 U.S.C. 103(a).
4. Claims 1-53 are pending. Claims 1, 24-25, 32, 42 and 46 are independent.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1-7, 10-16, 24-29, 32-33, 35, 42-44 and 46-49 are rejected under 35

U.S.C. 103(a) as being unpatentable over Hütsch et al (US Patent Application Publication No. 2001/0034771, filed Jan. 12, 2001, hereafter referred to as "Hütsch") in view of Alan Richmond, "HTML's META-tag: HTTP-EQUIV", Web Developer's <Virtual Library>, Oct. 12, 1999, pp. 1-3, hereafter referred to as "Richmond").

Regarding independent method claim 1, Hütsch discloses:

A method of incrementally rendering content in a content framework, comprising:

receiving a request for a portal page, wherein one or more portlets provide content for the portal page; ([0018] in context of Fig. 3A)

immediately returning a response message containing a first document the first document representing results from portlets which have acquired their content; ([0100] in context of Fig. 7, Fig. 8) and

..., if the first document does not represent results of all portlets. ([0127] – [0128])

However, Hütsch does not explicitly disclose:

programmatically generating a mechanism for delivering an updated document . . .

Richmond, though, discloses:

programmatically generating a mechanism for delivering an updated document . . . (p. 1 middle of page discussing code for HTTP-EQUIV = "Expires" and sentence stating "This can be used by caches to determine when to fetch a fresh copy of the associated document.")

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Richmond for the benefit of Hütsch, because to do so would allow a programmer to automatically refresh a document as taught by Richmond

in p. 1, middle of page discussing the HTTP-EQUIV = "Expires" attribute. These references were all applicable to the same field of endeavor, i.e., web page/service design.

Regarding claim 2, which is dependent upon claim 1, the limitations of claim 1 have been previously discussed.

However, Hütsch does not explicitly disclose:

wherein the programmatically generated mechanism comprises inclusion of a refresh trigger in the response message.

Richmond, though, discloses:

wherein the programmatically generated mechanism comprises inclusion of a refresh trigger in the response message. (p. 2 "Meta Refresh" section)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Richmond for the benefit of Hütsch, because to do so would allow a programmer to redirect a request for a URL as taught by Richmond in p. 2, "Meta Refresh" section. These references were all applicable to the same field of endeavor, i.e., web page/service design.

Regarding claim 3, which is dependent upon claim 2, the limitations of claim 2 have been previously discussed.

However, Hütsch does not explicitly disclose:

wherein the refresh trigger is a refresh header of the response message.

Richmond, though, discloses:

wherein the refresh trigger is a refresh header of the response message. (p. 2 "Meta Refresh" section)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Richmond for the benefit of Hütsch, because to do so would allow a programmer to redirect a request for a URL as taught by Richmond in p. 2, "Meta Refresh" section. These references were all applicable to the same field of endeavor, i.e., web page/service design.

Regarding claim 4, which is dependent upon claim 2, the limitations of claim 2 have been previously discussed.

However, Hütsch does not explicitly disclose:

wherein the refresh trigger is encoded using syntax of a markup language.

Richmond, though, discloses:

wherein the refresh trigger is encoded using syntax of a markup language. (p. 2 "Meta Refresh" section)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Richmond for the benefit of Hütsch, because to do so would allow a programmer to redirect a request for a URL as taught by Richmond in p.

Art Unit: 2176

2, "Meta Refresh" section. These references were all applicable to the same field of endeavor, i.e., web page/service design.

Regarding claim 5, which is dependent upon claim 4, the limitations of claim 4 have been previously discussed.

However, Hütsch does not explicitly disclose:

wherein the markup language is HTML ("Hypertext Markup Language").

Richmond, though, discloses:

wherein the markup language is HTML ("Hypertext Markup Language"). (p. 2 "Meta Refresh" section)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Richmond for the benefit of Hütsch, because to do so would allow a programmer to redirect a request for a URL as taught by Richmond in p. 2, "Meta Refresh" section. These references were all applicable to the same field of endeavor, i.e., web page/service design.

Regarding claim 6, which is dependent upon claim 5, the limitations of claim 5 have been previously discussed.

However, Hütsch does not explicitly disclose:

wherein the syntax composes a "META" tag using an "HTTP-EQUIV" attribute syntax.

Richmond, though, discloses:

wherein the syntax composes a "META" tag using an "HTTP-EQUIV" attribute syntax. (p. 2 "Meta Refresh" section)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Richmond for the benefit of Hütsch, because to do so would allow a programmer to redirect a request for a URL as taught by Richmond in p. 2, "Meta Refresh" section. These references were all applicable to the same field of endeavor, i.e., web page/service design.

Regarding claim 7, which is dependent upon claim 4, the limitations of claim 4 have been previously discussed.

Hütsch further discloses:

wherein the markup language is WML ("Wireless Markup Language"). ([0099])

Regarding claim 10, which is dependent upon claim 2, the limitations of claim 2 have been previously discussed.

However, Hütsch does not explicitly disclose:

wherein a value on the refresh trigger specifies a time before which the programmatically generated mechanism does not execute.

Richmond, though, discloses:

wherein a value on the refresh trigger specifies a time before which the programmatically generated mechanism does not execute. (p. 2 "Meta

Refresh" section, the META tag indicating that the URL is not invoked until 90 seconds have elapsed)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Richmond for the benefit of Hütsch, because to do so would allow a programmer to redirect a request for a URL as taught by Richmond in p. 2, "Meta Refresh" section. These references were all applicable to the same field of endeavor, i.e., web page/service design.

Regarding claim 11, which is dependent upon claim 2, the limitations of claim 2 have been previously discussed.

However, Hütsch does not explicitly disclose:

wherein a value on the refresh trigger is computed as a time after which a sender of the portal page request automatically invokes the delivery of the updated document.

Richmond, though, discloses:

wherein a value on the refresh trigger is computed as a time after which a sender of the portal page request automatically invokes the delivery of the updated document. (p. 2 "Meta Refresh" section, the META tag indicating that the URL is not invoked until 90 seconds have elapsed)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Richmond for the benefit of Hütsch, because to do so would allow a programmer to redirect a request for a URL as taught by Richmond in p. 2, "Meta Refresh" section. These references were all applicable to the same field of endeavor, i.e., web page/service design.

Regarding claim 12, which is dependent upon claim 2, the limitations of claim 2 have been previously discussed.

However, Hütsch does not explicitly disclose:

wherein a value on the refresh trigger is computed as a latest predicted completion time of a final one of the portlets which have not yet acquired their content.

Richmond, though, discloses:

wherein a value on the refresh trigger is computed as a latest predicted completion time of a final one of the portlets which have not yet acquired their content. (p. 2 "Meta Refresh" section, the META tag indicating that the URL is not invoked until 90 seconds have elapsed)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Richmond for the benefit of Hütsch, because to do so would allow a programmer to redirect a request for a URL as taught by Richmond in p. 2, "Meta Refresh" section. These references were all applicable to the same field of endeavor, i.e., web page/service design.

Regarding claim 13, which is dependent upon claim 12, the limitations of claim 12 have been previously discussed.

Hütsch further discloses:

wherein the value is determined by weighting actual fetch times of the portlets which have not yet acquired their content. ([0151 in context of [0156])

Regarding claim 14, which is dependent upon claim 12, the limitations of claim 12 have been previously discussed.

Hütsch further discloses:

wherein the value is determined by adding a constant value to a largest of weighted actual fetch times of the portlets which have not yet acquired their content. ([0151 in context of [0156])

Regarding claim 15, which is dependent upon claim 2, the limitations of claim 2 have been previously discussed.

Hütsch further discloses:

receiving the response message by a client from which the request for the portal page was sent; ([0100], esp. discussing transmission to the client, it being implicit that if one transmits to the client that the client may receive those transmissions)

rendering, by the client, the first document from the received response message; ([0100], esp. "for display") and

However, Hütsch does not explicitly disclose:

automatically sending a subsequent request for the portal page after waiting for a time specified by a value of the refresh trigger.

Richmond, though, discloses:

automatically sending a subsequent request for the portal page after waiting for a time specified by a value of the refresh trigger. (p. 2 "Meta Refresh" section, the META tag indicating that the URL is not invoked until 90 seconds [content value] have elapsed)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Richmond for the benefit of Hütsch, because to do so would allow a programmer to redirect a request for a URL as taught by Richmond in p. 2, "Meta Refresh" section. These references were all applicable to the same field of endeavor, i.e., web page/service design.

Regarding claim 16, which is dependent upon claim 2, the limitations of claim 2 have been previously discussed.

However, Hütsch does not explicitly disclose:

receiving a subsequent request for the portal page, the subsequent request having been automatically sent responsive to receiving the refresh trigger; and

returning a subsequent response comprising the updated document, responsive to receiving the subsequent request, the updated document being a subsequent version of the first document and representing results from portlets which have acquired their content thus far and which omits the refresh trigger only if all portlets have now acquired their content.

Richmond, though, discloses:

receiving a subsequent request for the portal page, the subsequent request having been automatically sent responsive to receiving the refresh trigger; (p. 2 "Meta Refresh" section, the META tag indicating that the URL is not invoked until 90 seconds [content value] have elapsed)and

returning a subsequent response comprising the updated document, responsive to receiving the subsequent request, the updated document being a subsequent version of the first document and representing results from portlets which have acquired their content thus far and which omits the refresh trigger only if all portlets have now acquired their content. (p. 2 "Meta Refresh" section, the META tag indicating that the URL is not invoked until 90 seconds [content value] have elapsed)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Richmond for the benefit of Hütsch, because to do so would allow a programmer to redirect a request for a URL as taught by Richmond in p. 2, "Meta Refresh" section. These references were all applicable to the same field of endeavor, i.e., web page/service design.

Regarding independent method claim 24, Hütsch discloses:

A method of incrementally rendering content in a content framework, comprising:

receiving a request for a portal page, wherein one or more portlets provide content for the portal page; ([0018] in context of Fig. 3A)
immediately returning a response message containing a first document, the first document representing results from portlets which have acquired their content; ([0100] in context of Fig. 7, Fig. 8) and
..., if the first document does not represent results of all portlets. ([0127) – [0128])

However, Hütsch does not explicitly disclose:

automatically delivering an updated document . . .

Richmond, though, discloses:

automatically delivering an updated document . . . (p. 1 middle of page discussing code for HTTP-EQUIV = "Expires" and sentence stating "This can be used by caches to determine when to fetch a fresh copy of the associated document.")

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Richmond for the benefit of Hütsch, because to do so would allow a programmer to automatically refresh a document as taught by Richmond in p. 1, middle of page discussing the HTTP-EQUIV = "Expires" attribute. These references were all applicable to the same field of endeavor, i.e., web page/service design.

Regarding independent method claim 25, Hütsch discloses:

A method of incrementally rendering content in a content framework, comprising:

receiving a request for a portal page frame, wherein one or more portlets provide content for the portal page frame; ([0018] in context of Fig. 3A)

immediately returning a response message containing a first mini-document, the first document representing results from portlets which have acquired their content; ([0100] in context of Fig. 7, Fig. 8) and

..., if the first document does not represent results of all portlets. ([0127] – [0128])

However, Hütsch does not explicitly disclose:

programmatically generating a mechanism for delivering an updated mini-document

Richmond, though, discloses:

programmatically generating a mechanism for delivering an updated mini-document (p. 1 middle of page discussing code for HTTP-EQUIV = "Expires" and sentence stating "This can be used by caches to determine when to fetch a fresh copy of the associated document.")

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Richmond for the benefit of Hütsch, because to do so would allow a programmer to automatically refresh a document as taught by Richmond in p. 1, middle of page discussing the HTTP-EQUIV = "Expires" attribute. These references were all applicable to the same field of endeavor, i.e., web page/service design.

Claims 26 and 43 are substantially similar to claim 3 and therefore likewise rejected.

Claims 27 and 44 are substantially similar to claim 4 and therefore likewise rejected.

Claim 28 incorporates the limitations of claims 5 and 6, and therefore is substantially similar to these claims and likewise rejected.

Claim 29 is substantially similar to claim 11 and therefore likewise rejected.

Regarding independent system claim 32, Hütsch discloses:

A system for incrementally rendering content in a content framework, comprising:

means for receiving a request for a portal page, wherein one or more portlets provide content for the portal page; ([0018] in context of Fig. 3A)

means for immediately returning a response message containing a first document, the first document representing results from portlets which have acquired their content; ([0100] in context of Fig. 7, Fig. 8) and

..., if the first document does not represent results of all portlets. ([0127) – [0128])

However, Hütsch does not explicitly disclose:

means for programmatically generating a mechanism for delivering an updated document

Richmond, though, discloses:

means for programmatically generating a mechanism for delivering an updated document (p. 1 middle of page discussing code for HTTP-EQUIV = “Expires” and sentence stating “This can be used by caches to determine when to fetch a fresh copy of the associated document.”)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Richmond for the benefit of Hütsch, because to do so would allow a programmer to automatically refresh a document as taught by Richmond in p. 1, middle of page discussing the HTTP-EQUIV = “Expires” attribute. These references were all applicable to the same field of endeavor, i.e., web page/service design.

Claims 33 and 47 are substantially similar to claim 2 and therefore likewise rejected.

Claims 35 and 49 are substantially similar to claim 16 and therefore likewise rejected.

Regarding independent system claim 42, Hütsch discloses:

A system for incrementally rendering content in a content framework, comprising:

means for receiving a request for a portal page frame, wherein one or more portlets provide content for the portal page frame; ([0018] in context of Fig. 3A)

means for immediately returning a response message containing a first mini-document, the first document representing results from portlets which have acquired their content; ([0100] in context of Fig. 7, Fig. 8) and

..., if the first document does not represent results of all portlets. ([0127) – [0128])

However, Hütsch does not explicitly disclose:

means for programmatically generating a mechanism for delivering an updated mini-document

Richmond, though, discloses:

means for programmatically generating a mechanism for delivering an updated mini-document (p. 1 middle of page discussing code for HTTP-EQUIV = "Expires" and sentence stating "This can be used by caches to determine when to fetch a fresh copy of the associated document.")

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Richmond for the benefit of Hütsch, because to do so would allow a programmer to automatically refresh a document as taught by Richmond in p. 1, middle of page discussing the HTTP-EQUIV = "Expires" attribute. These

references were all applicable to the same field of endeavor, i.e., web page/service design.

Regarding independent computer program product claim 46, Hütsch discloses:

A computer program product incrementally rendering content in a content framework, the computer program product embodied on one or more computer-readable media and comprising:

computer readable program code configured to receive a request for a portal page, wherein one or more portlets provide content for the portal page; ([0018] in context of Fig. 3A)

computer readable program code configured to immediately return a response message containing a first document, the first document representing results from portlets which have acquired their content; ([0100] in context of Fig. 7, Fig. 8) and

..., if the first document does not represent results of all portlets. ([0127) – [0128])

However, Hütsch does not explicitly disclose:

computer readable program code configured to programmatically generate a mechanism for delivering an updated document

Richmond, though, discloses:

computer readable program code configured to programmatically generate a mechanism for delivering an updated document (p. 1 middle of page discussing code for HTTP-EQUIV = "Expires" and sentence stating "This can be used by caches to determine when to fetch a fresh copy of the associated document.")

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Richmond for the benefit of Hütsch, because to do so

Art Unit: 2176

would allow a programmer to automatically refresh a document as taught by Richmond in p. 1, middle of page discussing the HTTP-EQUIV = "Expires" attribute. These references were all applicable to the same field of endeavor, i.e., web page/service design.

Claim 48 is substantially similar to claim 15 and therefore likewise rejected.

7. **Claims 8-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hütsch (US Patent No. 6,668,353, filed Mar. 25, 1999, hereafter referred to as "Hütsch") in view of Alan Richmond, "HTML's META-tag: HTTP-EQUIV", Web Developer's <Virtual Library>, Oct. 12, 1999, pp. 1-3 (hereafter referred to as "Richmond") and further in view of Morris (US Patent No. 6,453,361, filed Oct. 27, 2000, hereafter referred to as "Morris").**

Regarding claim 8, which is dependent upon claim 4, the limitations of claim 4 have been previously discussed.

However, Hütsch does not explicitly disclose:

wherein the markup language is i-mode format.

Morris, though, discloses:

wherein the markup language is i-mode format. (col. 4 lines 48-53, noting that cHTML is the actual markup language and that i-mode is a service, as per col. 2 lines 53-54)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Morris for the benefit of Hütsch in view of Richmond, because to do so would allow a user to communicate using a client device such as a cell phone as taught by Morris in col. 4 lines 47-53. These references were all applicable to the same field of endeavor, i.e., web page/service design.

Regarding claim 9, which is dependent upon claim 4, the limitations of claim 4 have been previously discussed.

However, Hütsch does not explicitly disclose:

wherein the markup language is HDML (“Handheld Device Markup Language”).

Morris, though, discloses:

wherein the markup language is HDML (“Handheld Device Markup Language”). (col. 4 lines 48-53)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Morris for the benefit of Hütsch in view of Richmond, because to do so would allow a user to communicate using a client device such as a cell phone as taught by Morris in col. 4 lines 47-53. These references were all applicable to the same field of endeavor, i.e., web page/service design.

8. **Claims 17-22, 30-31, 34, 36-40, 45, and 50-53 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hütsch (US Patent No. 6,668,353, filed Mar. 25, 1999, hereafter referred to as "Hütsch") in view of Alan Richmond, "HTML's META-tag: HTTP-EQUIV", Web Developer's <Virtual Library>, Oct. 12, 1999, pp. 1-3 (hereafter referred to as "Richmond") and further in view of Laura LeMay, SAMS Teach Yourself Web Publishing with HTML 4 in 21 Days, 2nd Edition, Sam's Publishing, Indianapolis, IN, © 2000 (hereafter referred to as "LeMay").**

Regarding claim 17, which is dependent upon claim 1, the limitations of claim 1 have been previously discussed.

However, Hütsch does not explicitly disclose:

wherein the programmatically generated mechanism comprises creating a multipart document, and wherein the first document is embedded in a first of the parts of the multipart document.

LeMay, though, discloses:

wherein the programmatically generated mechanism comprises creating a multipart document (p. 364, especially highlighted code at bottom of page), and wherein the first document is embedded in a first of the parts of the multipart document. (p. 365, Figure 12.10)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of LeMay for the benefit of Hütsch in view of Richmond, because to do so would allow a web publisher to display more than one HTML document, for instance, within a single browser as taught by LeMay in the p. 360

Art Unit: 2176

“Working with Frames” section, including Figure 12.7. These references were all applicable to the same field of endeavor, i.e., web page/service design.

Regarding claim 18, which is dependent upon claim 17, the limitations of claim 17 have been previously discussed.

However, Hütsch does not explicitly disclose:

wherein the first of the parts is preceded by a boundary string used to delimit parts of the multipart document.

LeMay, though, discloses:

wherein the first of the parts is preceded by a boundary string used to delimit parts of the multipart document. (p. 364, especially the highlighted code at bottom of page: <frameset rows=",*,*>)*

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of LeMay for the benefit of Hütsch in view of Richmond, because to do so would allow a web publisher to display more than one HTML document, for instance, within a single browser as taught by LeMay in the p. 360 “Working with Frames” section, including Figure 12.7. These references were all applicable to the same field of endeavor, i.e., web page/service design.

Regarding claim 19, which is dependent upon claim 18, the limitations of claim 18 have been previously discussed.

However, Hütsch does not explicitly disclose:

wherein the first of the parts is followed by a terminating boundary string if the first document represents results from all portlets.

LeMay, though, discloses:

wherein the first of the parts is followed by a terminating boundary string if the first document represents results from all portlets. (p. 364, especially the highlighted code at bottom of page: </frameset>)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of LeMay for the benefit of Hütsch in view of Richmond, because to do so would allow a web publisher to display more than one HTML document, for instance, within a single browser as taught by LeMay in the p. 360 “Working with Frames” section, including Figure 12.7. These references were all applicable to the same field of endeavor, i.e., web page/service design.

Regarding claim 20, which is dependent upon claim 17, the limitations of claim 17 have been previously discussed.

Hütsch further discloses:

receiving the response message by a client from which the request for the portal page was sent; ([0100], esp. discussing transmission to the client, it being implicit that if one transmits to the client that the client may receive those transmissions)

rendering, by the client, the first document from the first of the parts of the multipart document; ([0100], esp. “for display”)

receiving, by the client, subsequent parts of the multipart document, each of the subsequent parts comprising a revised version of the first document; ([0100], esp. discussing transmission to the client, it being implicit that if one transmits to the client that the client may receive those transmissions) and

rendering, by the client, the subsequent parts of the multipart document. ([0100], esp. “for display”)

Regarding claim 21, which is dependent upon claim 17, the limitations of claim 17 have been previously discussed.

However, Hütsch does not explicitly disclose:

detecting that one or more of the portlets which had not acquired their content when the first document was returned in the response message have now acquired their content; and
sending, responsive to the detecting step, a subsequent response message containing a revised version of the first document, the revised version representing results from the one or more portlets and being embedded in a subsequent part of the multipart document.

LeMay, though, discloses:

detecting that one or more of the portlets which had not acquired their content when the first document was returned in the response message have now acquired their content; (p. 2 “Meta Refresh” section, the META tag indicating that the URL is not invoked until 90 seconds [content value] have elapsed) and
sending, responsive to the detecting step, a subsequent response message containing a revised version of the first document, the revised version representing results from the one or more portlets and being embedded in a subsequent part of the multipart document. (p. 2 “Meta Refresh” section, the META tag indicating that the URL is not invoked until 90 seconds [content value] have elapsed)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Richmond for the benefit of Hütsch, because to do so would allow a programmer to redirect a request for a URL as taught by Richmond in p. 2, “Meta Refresh” section. These references were all applicable to the same field of endeavor, i.e., web page/service design.

Claim 22 incorporates the limitations of claims 18 and 19, and therefore is substantially similar to these claims and likewise rejected.

Claims 31, 37, 40 and 53 are substantially similar to claim 22 and therefore likewise rejected.

Claims 30, 36 and 45 are substantially similar to claim 17 and therefore likewise rejected.

Claims 38 and 51 are substantially similar to claim 20 and therefore likewise rejected.

Claims 39 and 52 are substantially similar to claim 21 and therefore likewise rejected.

Claim 50 incorporates the limitations of claims 17, 18 and 19, and therefore is substantially similar to these claims and likewise rejected.

Regarding claim 34, which is dependent upon claim 32, the limitations of claim 32 have been previously discussed.

Hütsch further discloses:

means for receiving the response message by a client from which the request for the portal page was sent; ([0100], esp. discussing

transmission to the client, it being implicit that if one transmits to the client that the client may receive those transmissions)

means for rendering, by the client, the first document from the received response message; ([0100], esp. "for display")

However, Hütsch does not explicitly disclose:

means for automatically sending a subsequent request for the portal page after waiting for a time specified by a value of the refresh trigger.

Richmond, though, discloses:

means for automatically sending a subsequent request for the portal page after waiting for a time specified by a value of the refresh trigger. (p. 2 "Meta Refresh" section, the META tag indicating that the URL is not invoked until 90 seconds [content value] have elapsed)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Richmond for the benefit of Hütsch, because to do so would allow a programmer to redirect a request for a URL as taught by Richmond in p. 2, "Meta Refresh" section. These references were all applicable to the same field of endeavor, i.e., web page/service design.

9. **Claims 23 and 41 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hütsch (US Patent No. 6,668,353, filed Mar. 25, 1999, hereafter referred to as "Hütsch") in view of Alan Richmond, "HTML's META-tag: HTTP-EQUIV", Web Developer's <Virtual Library>, Oct. 12, 1999, pp. 1-3 (hereafter referred to as "Richmond") and further in view of Kanefsky et al. (US Patent Application Publication**

No. 2002/0026500, provisionally filed Jun. 12, 2000, hereafter referred to as "Kanefsky").

Regarding claim 23, which is dependent upon claim 1, the limitations of claim 1 have been previously discussed.

However, Hütsch does not explicitly disclose:

wherein the programmatically generated mechanism comprises programmatically inserting a hyperlink into the first document, wherein the inserted hyperlink can be used to explicitly request delivery of the updated document.

Kanefsky, though, discloses:

wherein the programmatically generated mechanism comprises programmatically inserting a hyperlink into the first document, wherein the inserted hyperlink can be used to explicitly request delivery of the updated document. ([0062], especially "the application server ... modifies an existing URL ... to contain a link The user ... may select the URL")

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Kanefsky for the benefit of Hütsch in view of Richmond, because to do so would allow a server to perform relaying services to devices attached to a network as taught by Kanefsky in [0027]. These references were all applicable to the same field of endeavor, i.e., web page/service design.

Claim 41 is substantially similar to claim 23 and therefore likewise rejected.

Response to Arguments

10. Applicant's arguments have been fully considered but they are not persuasive.

The previous rejections of the claims under 35 USC 103(a):

The Office has withdrawn the previous rejections under 35 USC 103(a), as stated above.

Conclusion

11. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Non-patent Literature

Butler, David, et al., "OmniSearch: A method for searching dynamic content on the Web", ACM SIGMOD 2001, Santa Barbara, CA, May 21-24, 2001, p. 604 [ACM 1-58113-332-4/01/05].

Chen, Liren, et al., "WebMate: A Personal Agent for Browsing and Searching", Autonomous Agents '98, Minneapolis, MN, © 1998, pp. 132-139 [ACM 0-89791-983-1/98/5].

Yew, A., et al., "Customisable Off-Line Web Browsing with Mobile Software Agents", © 2000 IEEE, pp. 102-108 [0-7803-7133-X/00].

Musciano, Chuck, et al., HTML & XHTML: The Definitive Guide, 4th Edition, O'Reilly Publishing, Aug. 2000, pp. 1-7 (downloaded from: skaiste.elektalit/Books/O'Reilly/Bookshelfs/books/webdesign/xhtml/index.htm), ISBN: 0-596-00026-X.

Niederst, Jennifer, Web Design in a Nutshell, 2nd Edition, O'Reilly Publishing, Sep. 2001, pp. 1-7 (downloaded from: skaiste.elektalit/Books/O'Reilly/Bookshelfs/books/webdesign/wdesign/index.htm), ISBN: 0-596-00196-7.

US Patent Application Publications

Kahan et al	2002/0024536
Hackbarth et al	2002/0147777

US Patents

Rajan et al	6,871,220
Van Horne et al	6,460,084
Sathyanarayan et al	6,691,106
Le Pennec et al	6,920,488
Chipman et al	6,799,174
Johnson	6,871,197
Kumar et al	6,278,993

12. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Robert M Stevens whose telephone number is (571) 272-4102. The examiner can normally be reached on M-F 6:00 - 2:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Heather R. Herndon can be reached on (571) 272-4136. The current fax phone number for the organization where this application or proceeding is assigned is 703-872-9306. Additionally, the main number for Technology Center 2100 is (571) 272-2100.

Art Unit: 2176

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Robert Stevens
Art Unit 2176
Date: December 24, 2005

rms



**WILLIAM BASHORE
PRIMARY EXAMINER**

